GroupEnc: encoder with group loss for global structure preservation

David Novak[1,2][0000−0003−4574−9093], Sofie Van Gassen[1,2][0000−0002−7119−5330], and Yvan Saeys[1,2][0000−0002−0415−1506]

[1] Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Belgium
[2] Data Mining and Modeling for Biomedicine, Center for Inflammation Research, VIB-UGent, Belgium

Abstract. Recent advances in dimensionality reduction have achieved more accurate lower-dimensional embeddings of high-dimensional data. In addition to visualisation, these embeddings can be used for downstream processing, including batch effect correction, clustering, community detection or trajectory inference. We use the notion of structure preservation at both local and global levels to create a deep learning model, based on a variational autoencoder (VAE) and the stochastic quartet loss from the *SQuadMDS* algorithm. Our encoder model, called *GroupEnc*, uses a 'group loss' function to create embeddings with less global structure distortion than VAEs do, while keeping the model parametric and the architecture flexible. We validate our approach using 5 publicly available biological single-cell transcriptomic datasets. Employing R_{NX} curves for evaluation, we demonstrate consistently improved preservation of global structure over a VAE model.

Keywords: Dimensionality reduction · Autoencoders · Bioinformatics.

1 Introduction

Autoencoders (AEs) are neural networks which encode high-dimensional (HD) input data as a low-dimensional (LD) latent representation and decode this into a reconstruction of the input. In training, reconstruction error is minimised via back-propagation.

In the field of bioinformatics, we have seen impressive applications of autoencoders and variational autoencoders (VAEs; probabilistic models based on AEs) in dimensionality reduction (DR) for the purposes of visualisation [16, 4] and downstream data processing, including batch effect correction and cell population clustering [2, 3, 8]. This pertains to large and high-dimensional single-cell

This research was funded by the Flemish Government (AI Research Program) and by the FWO (Fonds Wetenschappelijk Onderzoek) and conducted as part of the FWO Strategic Basic research project 1S40421N.

datasets, which quantify biological features per cell in a tissue sample of interest. Examples of these methods include single-cell RNA sequencing (scRNA-seq), flow cytometry, mass cytometry (*CyTOF*) and CITE-seq.

Popular methods for DR of biological data, like t-SNE [17] and *UMAP* [12], are fundamentally aimed at preservation of local structures. For the purposes of inferring developmental trajectories or embedding and identifying outlier populations, the preservation of global-scale structures is desirable.

We introduce and evaluate *GroupEnc*: a stand-alone encoder module that optimises the *group loss*: a differentiable loss function that imposes a scale-agnostic structure-preserving constraint on the learned LD embedding. This is a modification of the stochastic quartet loss in *SQuadMDS* [9], applied in a deep learning context here. This results in a parametric model that can run on GPU. We achieve similar local structure preservation and better global structure preservation than a VAE model, as tested on 5 single-cell transcriptomic datasets. Compared to previously published alternative triplet-based loss functions proposed for VAEs [16, 1], the group loss does not require computation of a k-nearest-neighbour graph of the input data.

2 Method

We describe the methodology used to create LD embeddings of HD data and to evaluate them.

2.1 Model training

In an autoencoder architecture, HD input $\mathbf{X_{n\times d}} \in \mathcal{X}$ is encoded as LD representation $\mathbf{Z_{n\times w}} \in \mathcal{Z}$ (where $\mathcal{X} = \mathbb{R}^d, \mathcal{Z} = \mathbb{R}^w, w < d$) and reconstructed as an approximation $\mathbf{\hat{X}_{n\times d}}$.

The encoder $E_\Phi : \mathcal{X} \to \mathcal{Z}$ transforms $\mathbf{X}$ to $\mathbf{L}$, and the decoder $D_\theta : \mathcal{Z} \to \mathcal{X}$ transforms $\mathbf{L}$ to $\mathbf{\hat{X}}$. Parameters of the AE (encoder weights Φ and decoder weights Θ) are learned so as to reduce a reconstruction loss. In our baseline VAE model, we use the mean square error (MSE) as reconstruction loss.

In a VAE, the latent representation $\mathbf{L}$ is sampled from a distribution $\mathcal{D}$ in latent space. The encoder and decoder networks are probabilistic, and an extra term quantifying the Kullback-Leibler (KL) divergence between $\mathcal{D}$ and a latent prior (isotropic Gaussian distribution) is used as an additional loss term during training.

In contrast, our current *GroupEnc* model only consists of a variational encoder and sampler (without a decoder), trained to minimise a *group loss* along with the KL divergence from prior. The group loss adapts the notion of the quartet loss function, computed using quartet-normalised distances between original and embedded points, from *SQuadMDS* [9]. The normalised distances are used to calculate a differentiable cost function per each randomly drawn quartet of points. We denote Euclidean distances between any HD input points or LD embedded points indexed i and j as δ_{ij} and d_{ij}, respectively. To compute a

group-normalised distance between two points in the same group (for a quartet, quintet, sextet, etc.), we use all pairwise distances within that group. For HD and LD points, respectively, we get group-normalised distance formulas

$$\delta_{ij}^{\text{norm}} = \frac{\delta_{ij}}{\sum_{a=1}^{\gamma-1} \sum_{b=a+1}^{\gamma} \delta_{ab}} \tag{1}$$

$$d_{ij}^{\text{norm}} = \frac{d_{ij}}{\sum_{a=1}^{\gamma-1} \sum_{b=a+1}^{\gamma} d_{ab}} \tag{2}$$

where γ is the number of points in each group.

The difference in group-normalised distances in HD and LD, which ought to be minimised, is used to calculate the cost function

$$g = \sum_{a=1}^{\gamma-1} \sum_{b=a+1}^{\gamma} (\delta_{ab}^{\text{norm}} - d_{ab}^{\text{norm}})^2 \tag{3}$$

of a group (a *group cost*). This is visualised in Figure 1.

The *GroupEnc* model is trained on shuffled batches of input data using the *Adam* optimiser. Partitioning of points into groups is done dynamically at the batch level (a new partition is made of each training batch when it is drawn from the data). The size of the groups (γ) is specified as a hyperparameter. The group loss value per each point i in the training batch is assigned as the cost value of the group for which i is the first point, and the group loss term per batch is averaged across the batch.

Therefore, *GroupEnc* imposes a constraint on the latent distribution $\mathcal{D}$ instead of reconstruction loss to compute weight updates.

2.2 Dimensionality reduction quality assessment

To assess structure preservation (SP) in an embedding, we use the R_{NX} curve, a previously proposed quality assessment metric [10]. This curve quantifies the overlap between ordering of neighbours to a reference point in HD versus in LD for all neighbourhood sizes, from 1 to $(N-1)$ (with sample size N), averaged across all reference points.

To compute this, we denote neighbourhood ranks of a point j (neighbour) with respect to a point i (reference point) as ρ_{ij} and r_{ij} in HD and in LD, respectively. Non-self neighbourhoods of HD and LD points, respectively, are then denoted as $\nu_i^K = \{j : 1 \leq \rho_{ij} \leq K\}$ and $n_i^K = \{j : 1 \leq r_{ij} \leq K\}$ for neighbourhood size K. For dataset size N, the Q_{NX} value for a specific value of K is calculated as

$$Q_{\text{NX}}(K) = \frac{1}{KN} \sum_{i=1}^{N} |\nu_i^K \cap n_i^K| \tag{4}$$

To obtain the full Q_{NX} curve, we calculate this score for K from 1 to $(N-1)$.

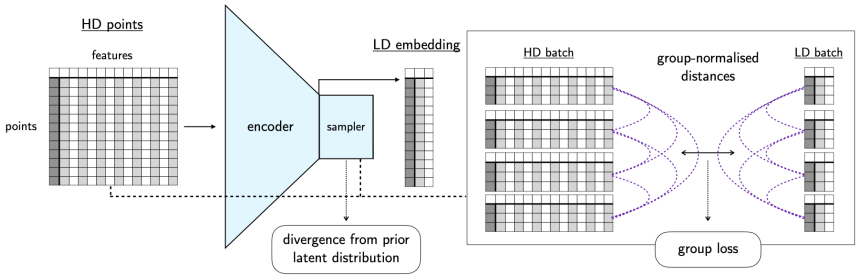

Fig. 1. A schematic illustration of *GroupEnc* training and inference is shown. In training, a batch of high-dimensional points is used as input to the encoder. Parameters of the model are adjusted in each pass via back-propagation to minimise the group loss, which quantifies divergence in relative distances within randomly assigned groups of points in a batch of input points versus its embedding. The trained encoder then outputs a low-dimensional embeddding of the input.

It turns out that a random embedding results in $Q_{\mathrm{NX}}(K) \approx \frac{K}{N-1}$. R_{NX}, as opposed to Q_{NX}, corrects for chance, and is computed as

$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K} \tag{5}$$

We quantify SP as the area-under-curve (AUC) for an R_{NX} curve of an embedding of interest. Specifically, Local SP is the AUC of the curve where neighbourhood size (K) is re-scaled logarithmically (lnK is used), to up-weight local neighbourhoods while not setting a hard cut-off for local versus global. Moreover, Global SP is the AUC with a linear scale for K, therefore without the emphasis on local neighbourhoods of the reference points. In both cases, a higher SP score is better.

3 Results

We compare a VAE (trained to minimise reconstruction error and KL-divergence from prior) and a *GroupEnc* model (encoder-only, trained to minimise group loss)*.

We tested structure preservation (SP) in embeddings of dimensionality 2, 5 and 10, with different values of hyperparameter γ (group size), looking at Local and Global SP separately.

*The encoder module, in both cases, consisted of layers sized $(32, 64, 128, 32)$ and the VAE decoder module consisted of layers sized $(32, 128, 64, 32)$. The *Adam* optimiser with a learning rate of 0.001 was used for 500 epochs of training with batch size of 512.

We use 5 single-cell RNA-sequencing (scRNA-seq) datasets [11, 5, 15, 20, 21], comprising high-dimensional feature vectors describing the identity of single biological cells in a tissue sample of interest. These features are levels of transcription of labelled genes. The datasets are listed in Table 3.

Local SP and Global SP scores are summarised in Figure 2 and shown in Tables 2 and 3 in full. Time required to train each model can be found in Table 4, with a single node of a GPU cluster (*16-core Intel Xeon Gold 6242* processor with *NVIDIA Volta V100* GPU) with 16 GB of usable RAM made available each time. 5 runs (with different random seeds) were run to collect the scores.

For the Farrell dataset, we also plot the 2-dimensional embeddings from both models and label individual embedded points using annotation provided by the authors (Figure 3). The labels are ordered and correspond to developmental stages of cells in zebrafish embryogenesis. This is to show the developmental gradient is more apparent in the *GroupEnc* embedding.

The results in Tables 2 and 3 show that, intuitively, both local and global structures in terms of neighbour ranks are preserved worse with decreased dimensionality of the embedding, and this holds across all tested datasets and models (VAE and *GroupEnc* with group sizes of 4, 5 and 6). Futhermore, results in Table 2 show that the VAE model generally outperforms the *GroupEnc* models when it comes to Local SP. However, per Table 3 we see consistently better Global SP for *GroupEnc*, concordant with the scale-agnostic nature of the group loss that *GroupEnc* optimises. This apparent trade-off between the two models is also captured clearly in Figure 2. We also see that differences between *GroupEnc* models with different group sizes are not significant.

We invoke Figure 3 to show a small case study regarding better preservation of a known biological developmental gradient by *GroupEnc* in comparison to a VAE. Based on this and the objective Local and Global SP results, we conclude that the scale-agnostic group loss term for learning a lower-dimensional embedding using a variational encoder does more to preserve the global structure of input data than a VAE. At the same time, the continuous nature of the encoding, the fact that the model remains parametric (thus allowing for transforming new data after training) and the possibility to use GPU acceleration are useful properties of *GroupEnc* in the context of working with large biological datasets.

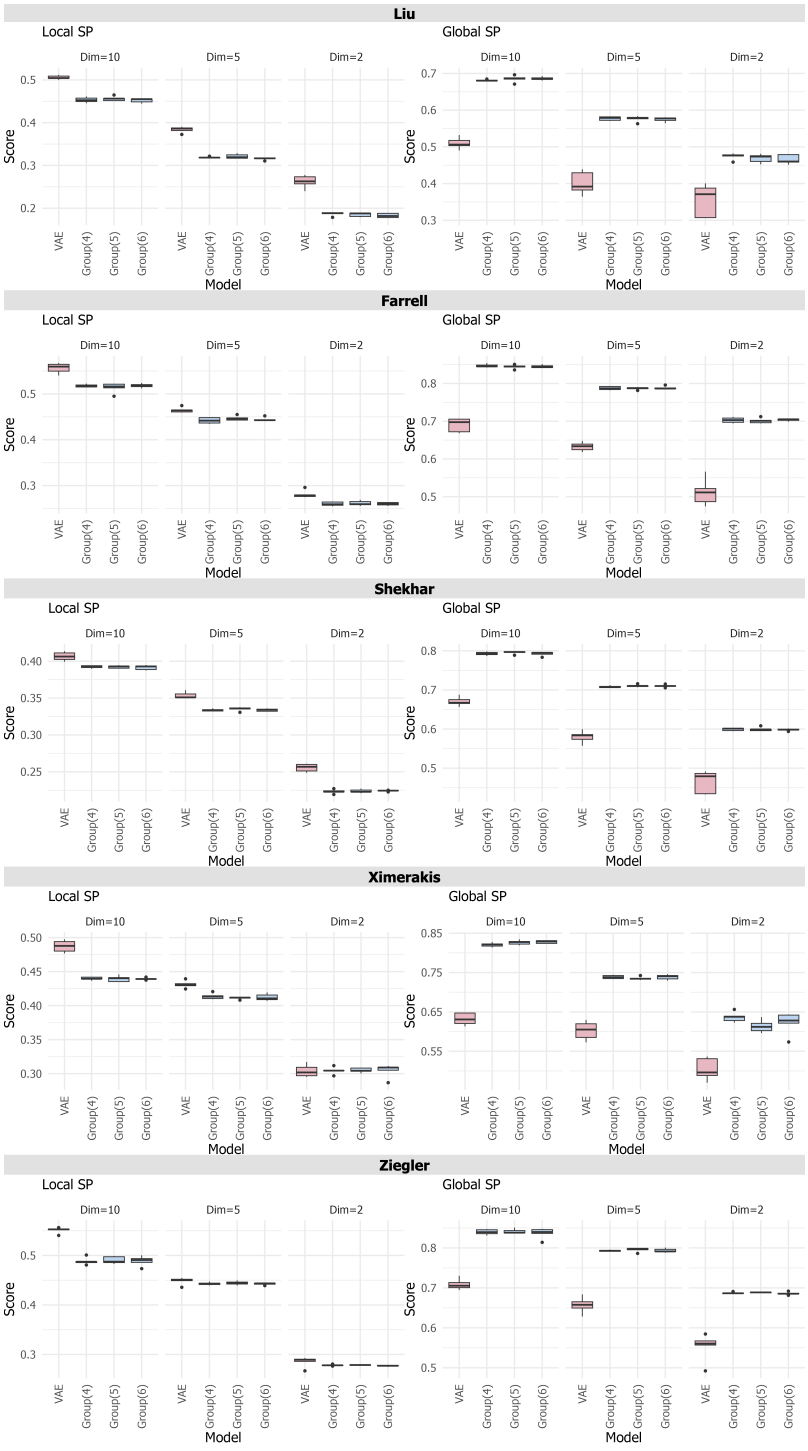

Fig. 2. Boxplots of Local and Global SP scores across 5 runs for embeddings of each dataset, obtained from VAE and *GroupEnc* model for different group size (γ) values (written as 'Group(γ)'). Subplots are sorted by target dimensionalities (columns).

Dataset name	Biological source	Feature set	Number of samples
Ziegler	human nasopharynx	32,871 genes	32,588 cells
Shekhar	mouse retina	24,904 genes	44,994 cells
Ximerakis	mouse brain	14,699 genes	37,069 cells
Farrell	zebrafish embryos	17,239 genes	38,731 cells
Liu	mouse brain	17,482 genes	26,187 cells

Table 1. Datasets used for DR benchmark and their brief descriptions.

Dim	Model	Dataset				
		Liu	Farrell	Shekhar	Ximerakis	Ziegler
$10d$	VAE	0.506±0.005	0.556±0.011	0.407±0.006	0.487±0.009	0.551±0.006
	GroupEnc ($\gamma = 4$)	0.453±0.006	0.518±0.003	0.392±0.002	0.440±0.002	0.488±0.007
	GroupEnc ($\gamma = 5$)	0.456±0.006	0.513±0.011	0.392±0.002	0.439±0.005	0.490±0.007
	GroupEnc ($\gamma = 6$)	0.452±0.005	0.518±0.004	0.391±0.003	0.439±0.002	0.489±0.010
$5d$	VAE	0.384±0.007	0.465±0.006	0.354±0.004	0.431±0.005	0.448±0.007
	GroupEnc ($\gamma = 4$)	0.318±0.002	0.442±0.007	0.334±0.002	0.414±0.004	0.443±0.003
	GroupEnc ($\gamma = 5$)	0.321±0.005	0.447±0.005	0.335±0.003	0.411±0.002	0.444±0.004
	GroupEnc ($\gamma = 6$)	0.316±0.003	0.444±0.004	0.334±0.002	0.412±0.005	0.443±0.002
$2d$	VAE	0.262±0.015	0.281±0.009	0.256±0.005	0.304±0.009	0.285±0.010
	GroupEnc ($\gamma = 4$)	0.187±0.005	0.260±0.005	0.223±0.003	0.304±0.005	0.278±0.001
	GroupEnc ($\gamma = 5$)	0.185±0.005	0.262±0.005	0.224±0.002	0.305±0.003	0.278±0.001
	GroupEnc ($\gamma = 6$)	0.183±0.006	0.260±0.003	0.224±0.001	0.304±0.010	0.277±0.001

Table 2. Local SP for 5 datasets, 3 embedding dimensionalities ('Dim') and 4 models (VAE and *GroupEnc* with group size γ of 4, 5 and 6). Mean and standard deviation are shown.

Dim	Model	Dataset				
		Liu	Farrell	Shekhar	Ximerakis	Ziegler
$10d$	VAE	0.510±0.016	0.690±0.019	0.670±0.012	0.632±0.015	0.709±0.014
	GroupEnc ($\gamma = 4$)	0.681±0.002	0.847±0.005	0.793±0.005	0.820±0.005	0.840±0.007
	GroupEnc ($\gamma = 5$)	0.685±0.009	0.844±0.005	0.796±0.004	0.826±0.006	0.841±0.006
	GroupEnc ($\gamma = 6$)	0.686±0.005	0.845±0.004	0.793±0.005	0.828±0.004	0.837±0.014
$5d$	VAE	0.401±0.032	0.633±0.012	0.580±0.016	0.602±0.024	0.657±0.020
	GroupEnc ($\gamma = 4$)	0.577±0.005	0.788±0.005	0.708±0.003	0.739±0.005	0.793±0.002
	GroupEnc ($\gamma = 5$)	0.576±0.008	0.787±0.003	0.710±0.003	0.735±0.004	0.795±0.005
	GroupEnc ($\gamma = 6$)	0.575±0.007	0.788±0.004	0.710±0.003	0.738±0.007	0.793±0.005
$2d$	VAE	0.355±0.045	0.512±0.036	0.465±0.029	0.504±0.028	0.552±0.035
	GroupEnc ($\gamma = 4$)	0.474±0.009	0.703±0.007	0.599±0.003	0.637±0.013	0.687±0.002
	GroupEnc ($\gamma = 5$)	0.469±0.012	0.701±0.007	0.599±0.005	0.614±0.016	0.689±0.001
	GroupEnc ($\gamma = 6$)	0.466±0.013	0.704±0.003	0.598±0.003	0.622±0.028	0.686±0.004

Table 3. Global SP for 5 datasets, 3 embedding dimensionalities ('Dim') and 4 models (VAE and *GroupEnc* with group size γ of 4, 5 and 6). Mean and standard deviation are shown.

Dim	Model	Dataset				
		Campbell	Farrell	Shekhar	Ximerakis	Ziegler
$10d$	VAE	89.7±9.4	126.2±15.9	138.2±15.8	116.4±11.2	103.9±11.3
	GroupEnc ($\gamma = 4$)	151.6±18.2	221.8±19.1	263.0±35.2	211.6±20.3	187.9±24.0
	GroupEnc ($\gamma = 5$)	150.2±15.5	228.7±27.0	252.1±25.6	222.9±23.2	172.9±5.3
	GroupEnc ($\gamma = 6$)	156.7±20.0	221.9±23.8	252.6±28.6	213.1±21.4	189.5±22.1
$5d$	VAE	78.9±3.5	121.9±10.8	137.8±15.4	120.1±12.9	100.2±6.7
	GroupEnc ($\gamma = 4$)	155.9±20.0	227.2±29.2	258.6±34.7	207.2±24.0	179.9±17.4
	GroupEnc ($\gamma = 5$)	148.5±17.1	226.7±28.9	249.9±30.2	207.8±24.4	177.3±18.3
	GroupEnc ($\gamma = 6$)	150.0±21.0	227.2±28.2	251.3±26.4	218.1±27.8	178.1±18.1
$2d$	VAE	85.4±1.6	123.3±10.7	149.5±12.9	117.1±12.2	100.1±11.1
	GroupEnc ($\gamma = 4$)	151.2±2.2	226.8±27.3	282.0±27.1	209.0±21.3	181.6±18.0
	GroupEnc ($\gamma = 5$)	159.7±16.0	217.3±24.0	255.6±23.5	209.0±22.7	179.1±17.3
	GroupEnc ($\gamma = 6$)	154.4±13.7	218.0±24.9	280.1±31.2	218.0±30.0	180.7±16.4

Table 4. Model training time in seconds across 5 datasets, 3 embedding dimensionalities ('Dim') and 4 models (VAE and *GroupEnc* with group size γ of 4, 5 and 6). Mean and standard deviation are shown.

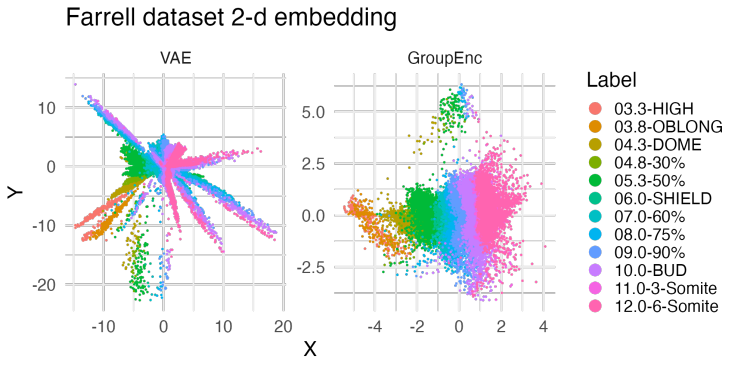

Fig. 3. 2-dimensional embeddings of the Farrell dataset obtained using VAE and *GroupEnc* ($\gamma = 4$) with colour labels according to labelled developmental stages of embedded cells.

4 Discussion

Faithful reconstructions of global relationships in lower-dimensional embeddings are of interest for purposes of visualisation, as well as the potential for downstream processing of data. We set out to design a deep learning model that uses a loss function for scale-agnostic preservation of randomly sampled structures [9].

We have done this to demonstrate the improvement in global structure preservation (versus VAE) via this loss function and that it can be used in a deep learning context, which has the advantage of providing a parametric model to be trained on a subset of data and used to transform new samples.

The use of geometric priors (similarity matrices, topological priors) with VAEs for dimensionality reduction [18, 8] is another promising avenue of research in analyses of high-dimensional datasets. With data that is high-dimensional and noisy by its nature (of which biological single-cell data is an instance), feature engineering by the means of constructing such lower-dimensional embeddings can help extract more salient information about the differential expression of genes in cells, continuous developmental gradients or batch effects between cohorts of samples.

In general, preserving global structures, as opposed to constraining the optimisation process to local structure preservation (as in t-SNE [17] or *UMAP* [12]) can prove beneficial for analysing hierarchical relationships, developmental gradients and pathways.

Our future work in dimensionality reduction of biological data will focus on effective reconstruction of trajectories, tackling noise and an extended range of evaluation metrics, both unsupervised and supervised.

To contextualise the new method, a systematic comparison to contractive autoencoders [13] and the recently proposed isometric autoencoders [7], encompassing a theoretical comparison as well as a benchmark, will be beneficial.

5 Code availability

We make a TensorFlow implementation of *GroupEnc*, including Bash scripts for generating benchmarking jobs (on Slurm) with custom datasets, available at `github.com/saeyslab/GroupEnc`.

5.1 Data availability

We downloaded the Shekhar and Liu datasets via the *scRNAseq* R package [14] using the functions `ShekharRetinaData` and `LiuBrainData` and converted them to `AnnData` objects using the *scDIOR* [6] packages for R/Python interoperability. Other datasets come from the *Single Cell Portal*[*] and are accessible using the following accession numbers.

- Farrell: `SCP162`
- Ximerakis: `SCP263`
- Ziegler: `SCP1289`
- Liu: `SCP2161`

[*] `https://singlecell.broadinstitute.org/single_cell`

5.2 Data pre-processing

We used the *scanpy* package version 1.9.1 [19] for data pre-processing. We applied the following Python code for scaling, normalisation and principal component analysis (PCA) prior to running the DR algorithms:

```
import scanpy as sc
hd = sc.pp.normalize_per_cell(X, copy=True) # assume X is count matrix
                                            # (numpy.ndarray)
hd = sc.pp.log1p(hd, copy=True)
hd = sc.pp.scale(hd, max_value=10, copy=True)
data = sc.tl.pca(hd, svd_solver='arpack', n_comps=50, copy=True)
```

The Farrell dataset was an exception, where already scaled data was used, and only the PCA step remained.

References

1. E. Amid and M. K. Warmuth. TriMap: Large-scale Dimensionality Reduction Using Triplets, Mar. 2022. arXiv:1910.00204 [cs, stat].
2. M. Amodio, D. van Dijk, K. Srinivasan, W. S. Chen, H. Mohsen, K. R. Moon, A. Campbell, Y. Zhao, X. Wang, M. Venkataswamy, A. Desai, V. Ravi, P. Kumar, R. Montgomery, G. Wolf, and S. Krishnaswamy. Exploring single-cell data with deep multitasking neural networks. *Nature Methods*, 16(11):1139–1145, Nov. 2019.
3. L. Chen, W. Wang, Y. Zhai, and M. Deng. Deep soft K-means clustering with self-training for single-cell RNA sequence data. *NAR Genomics and Bioinformatics*, 2(2):lqaa039, June 2020.
4. J. Ding, A. Condon, and S. P. Shah. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Communications*, 9(1):2002, May 2018.
5. J. A. Farrell, Y. Wang, S. J. Riesenfeld, K. Shekhar, A. Regev, and A. F. Schier. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science*, 360(6392):eaar3131, June 2018.
6. H. Feng, L. Lin, and J. Chen. scDIOR: single cell RNA-seq data IO software. *BMC Bioinformatics*, 23(1):16, Dec. 2022.
7. A. Gropp, M. Atzmon, and Y. Lipman. Isometric autoencoders, 2020.
8. A. Kopf, V. Fortuin, V. R. Somnath, and M. Claassen. Mixture-of-Experts Variational Autoencoder for clustering and generating from similarity-based representations on single cell data. *PLOS Computational Biology*, 17(6):e1009086, June 2021.
9. P. Lambert, C. de Bodt, M. Verleysen, and J. A. Lee. Stochastic quartet approach for fast multidimensional scaling. pages 417–422, 2021.
10. J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, Dec. 2015.
11. Y. Liu, E. L. Savier, V. J. DePiero, C. Chen, D. C. Schwalbe, R.-J. Abraham-Fan, H. Chen, J. N. Campbell, and J. Cang. Mapping visual functions onto molecular cell types in the mouse superior colliculus. *Neuron*, 111(12):1876–1886.e5, 2023.

12. L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, Sept. 2020. arXiv:1802.03426 [cs, stat].
13. S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 833–840, Madison, WI, USA, 2011. Omnipress.
14. D. Risso and M. Cole. *scRNAseq: Collection of Public Single-Cell RNA-Seq Datasets*, 2021. R package version 2.8.0.
15. K. Shekhar, S. W. Lapan, I. E. Whitney, N. M. Tran, E. Z. Macosko, M. Kowalczyk, X. Adiconis, J. Z. Levin, J. Nemesh, M. Goldman, S. A. McCarroll, C. L. Cepko, A. Regev, and J. R. Sanes. Comprehensive Classification of Retinal Bipolar Neurons by Single-Cell Transcriptomics. *Cell*, 166(5):1308–1323.e30, Aug. 2016.
16. B. Szubert, J. E. Cole, C. Monaco, and I. Drozdov. Structure-preserving visualisation of high dimensional single-cell datasets. *Scientific Reports*, 9(1):8914, June 2019.
17. L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.
18. R. Vandaele, B. Kang, J. Lijffijt, T. De Bie, and Y. Saeys. Topologically Regularized Data Embeddings, Mar. 2022. arXiv:2110.09193 [cs, stat].
19. F. A. Wolf, P. Angerer, and F. J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, Dec. 2018.
20. M. Ximerakis, S. L. Lipnick, B. T. Innes, S. K. Simmons, X. Adiconis, D. Dionne, B. A. Mayweather, L. Nguyen, Z. Niziolek, C. Ozek, V. L. Butty, R. Isserlin, S. M. Buchanan, S. S. Levine, A. Regev, G. D. Bader, J. Z. Levin, and L. L. Rubin. Single-cell transcriptomic profiling of the aging mouse brain. *Nature Neuroscience*, 22(10):1696–1708, Oct. 2019.
21. C. G. Ziegler, V. N. Miao, A. H. Owings, A. W. Navia, Y. Tang, J. D. Bromley, P. Lotfy, M. Sloan, H. Laird, H. B. Williams, M. George, R. S. Drake, T. Christian, A. Parker, C. B. Sindel, M. W. Burger, Y. Pride, M. Hasan, G. E. Abraham, M. Senitko, T. O. Robinson, A. K. Shalek, S. C. Glover, B. H. Horwitz, and J. Ordovas-Montanes. Impaired local intrinsic immunity to SARS-CoV-2 infection in severe COVID-19. *Cell*, 184(18):4713–4733.e22, 2021.